A Design of Overlay Anonymous Multicast Protocol

Li Xiao[1], Xiaomei Liu[1], Wenjun Gu[2], Dong Xuan[2], Yunhao Liu[3]

[1]Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA
[2]Dept. of Computer Science and Engineering, Ohio State University, Columbus, OH 43210, USA
[3]Dept. of Computer Science, Hong Kong Univ. of Science and Technology, Kowloon, Hong Kong

Abstract

Multicast services are demanded by a variety of applications. Many applications require anonymity during their communication. However, there has been very little work on anonymous multicasting and such services are not available yet. Since there are fundamental differences between multicast and unicast, the solutions proposed for anonymity in unicast communications cannot be directly applied to multicast applications. In this paper we define the anonymous multicast system, and propose a mutual anonymous multicast (MAM) protocol including the design of a unicast mutual anonymity protocol and construction and optimization of an anonymous multicast tree. MAM is self organizing and completely distributed. We define the attack model in an anonymous multicast system and analyze the anonymity degree. We also evaluate the performance of MAM by simulations.

1. Introduction

It is well known that multicast services are demanded by a variety of applications, e.g., Video conferencing, Internet based education, NASA TV, software updates etc. However the inability of the Internet to support multicast is patent. To alleviate this, multicast services on overlay networks have been proposed [3, 5, 6, 9, 10, 20, 22, 27, 29, 31, 33] and implemented [8] by the research community. The salient features of overlays include ease of deployment and flexibility. We envisage that in the near future, a wide variety of applications will be able to enjoy multicast services on overlay networks. Apart from commercial applications, we believe that government and military organizations will also use such services due to the several advantages multicast has to offer.

It follows to expect that, as multicast services continue to be deployed, existing and future multicast applications will also demand the security features that unicast communications have. Security in multicast communication has been addressed in [4, 19, 21]. Most of the work here focuses on authentication of the senders and receivers and the effi-cient distribution of the keys to all legal group members and exclusion of members leaving the group. Our focus here is providing anonymity in multicast communications. Anonymity is an important component of security and is demanded by many applications. Some of them are: Critical multicast services like military, emergency applications, where strategic information and critical updates are transmitted to multiple destinations needing anonymity from external observers. Multi-party video conferencing applications carrying classified information will need anonymity from external observers and other members in the group. Large business organizations may have to multicast database updates to many sites for synchronization, and such applications will demand anonymity from rival organizations.

Solutions proposed for anonymity in unicast communications can not be directly applied to multicast applications. The fundamental difference between multicast and unicast is the concept of a group in multicast. Due to the correlation among nodes, there are special challenges in achieving anonymity in multicast: (1) the anonymity semantics in multicast are different from those in unicast. For sender anonymity, the sender needs to hide not only from one receiver, but from a subset of, or all the receivers. In receiver anonymity, the receiver may need to hide not only from the sender, but also from other receiver(s). There is a special issue in anonymity in multicast called group anonymity, where the presence of the group is not disclosed to outsiders. (2) Multicast services naturally need the existence of a tree. Exposing this tree itself will compromise the degree of anonymity. In contrast, in unicast, the path from a sender to a receiver is much easier to hide. (3) Membership management is a challenging issue in multicast. Member joining and leaving makes anonymity difficult. (4) There are other inherent challenges for secure multicast services such as group key management etc.

In this paper, we use an overlay solution and propose the mutual anonymous multicast (MAM) protocol, including the design of a unicast mutual anonymity protocol and construction and optimization of an anonymous multicast tree. MAM is self organizing and completely distributed. The main contributions of this paper are as follows.

1. We define different types of anonymity in an anonymous multicast system, and show the rational of our focus on multicast mutual anonymity.

2. We propose the MAM protocol, and address the critical issues in this protocol, which include an efficient and robust unicast initiator anonymity protocol, an efficient unicast mutual anonymity protocol, and an effective anonymous multicast construction approach. The self-organized and completely distributed design of MAM can efficiently realize mutual anonymity in overlay multicast systems.

3. We define the attack model to an anonymous multicast system, and theoretically analyze the anonymity degree of the MAM protocol.

4. By an extended simulation study, we show the effectiveness of MAM in a dynamic environment.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the MAM protocol design. Section 4 analyzes anonymity degree. Section 5 evaluates the performance of the MAM protocol. We conclude the work in Section 6.

2. Related Work

We introduce the related work to this research here. We will focus the work on (1) basic multicast (no security and anonymity), (2) anonymous unicast, and (3) multicast anonymity.

2.1 Basic Multicast

Originally, multicast research was focused on network-layer. However, no real multicast service has been provided at the network layer. The main issues in network-layer multicast are deployment and scalability issues. Recently, the focus has been moved up to application-layer multicast [3, 5, 6, 9, 10, 20, 22, 27, 29, 31, 33]. It is well believed that application-layer solutions provide much flexibility in design and implementation. Overlays are increasingly being used to deploy network services for different applications since they have the advantages of being easy to implement and flexible in adapting to dynamic underlying networks[22]. A list of overlay applications is given in [22], such as file sharing and network-embedded storage, content distribution networks, routing and multicast overlays, QoS overlays, scalable object location, scalable event propagation, wireless and mobile systems, and cluster-based overlays constructed among sensor nodes [18].

2.2 Anonymous Unicast

Some work has been reported on overlay anonymous unicast, such as Tarzan [12] and our work [32]. The essential techniques to achieve unicast anonymity can be classified into the following categories: routing, addressing, layered encryption, and traffic covering. In the routing approach, there can be either indirect forwarding, i.e. the use of intermediate nodes (forwarders) to hide correlation between the sender and the receiver [7, 28], or flooding [26]. In addressing, it can be implicit, i.e. the address contains no information either on the actual location of the addressee or on the physical reachability of the addressee [14, 15], or explicit, i.e. the address contains information that can be used in a straightforward manner to route a message to the addressee [30]. Layered encryption is often used in anonymity protocols [7, 28]. Traffic covering can prevent traffic timing analysis [13, 17]. These techniques often work together to achieve anonymity. For example, indirect forwarding needs layered encryption to encrypt the identities of forwarders. Flooding needs implicit address.

There are two ways to choose forwarders in the indirect forwarding approaches. It can be in a centralized fashion, such as Onion [28], or a distributed fashion, such as Crowds [23] and Tor [11]. In the centralized fashion, some centers (maybe the sender or receiver) choose the whole list of forwarders and use layered encryption techniques to encrypt them. The list of forwarders will be piggybacked in the message. The problem is the center needs to know the global network information, which is not scalable in a large scale network. In the distributed fashion, during the message forwarding, the next-hop forwarder is decided by the current forwarder (there are certainly some variations). The mechanism is scalable and can be applied in sender anonymity. However, it is hard, if not impossible for the latter to be used in receiver anonymity.

2.3 Anonymous Multicast

Little work [16, 30] has been reported on anonymous multicasting. Anonymous multicast communication service is not available yet. Authors in [30] proposed the use of a proxy, called SAM server, to hide some receivers. The main idea is first to add a SAM server as a normal node into a multicast tree, then attach receivers to the SAM server so that they are hidden by the server from other members. The concept of SAM server is a kind of extension to proxy or mixer in unicast. There are some drawbacks of this system. If there are multiple receivers attached to a SAM server, there exists another multicast anonymity problem among these receivers. The SAM server can be a target of attack. It however should also be trusted. Some types of multicast anonymity have not been addressed, such as multicast mutual anonymity and multicast group anonymity.

3. MAM Protocol

In this section, we define multicast anonymity and present the design of MAM protocol.

3.1 Definition of Multicast Mutual Anonymity

We assume every node could be a sender and a receiver in the service. Nodes that are neither senders nor receivers are called outsiders to the group.

Definition 1: **Multicast mutual anonymity**. Here a set of members desire to be hidden from others. Members in such a set need to achieve mutual anonymity from each other. Such a set can be a pair, such as the sender and a receiver; or one receiver and another receiver. The set also can be multiple members and may even include all members (i.e. complete anonymity).

Multicast mutual anonymity can cover multicast sender anonymity (hide the sender's identity) and receiver anonymity (hide one or more receivers' identities). Of course multicast sender anonymity and receiver anonymity can be achieved by simpler protocols than that for multicast mutual anonymity. *Multicast mutual anonymity is the focus of this paper.* Another type of multicast anonymity is group anonymity, which hides the existence of a multicast group session from all outsiders. Traffic covering approaches can be used to achieve multicast group anonymity, which is out of the scope of this paper.

We define three types of nodes in a mutual anonymous multicast system.

(1) *Anonymous member nodes*, **AM** nodes in short, are the member nodes whose identities need to be hidden from all member/non-member nodes.

(2) *Non-anonymous member nodes*, **NM** nodes in short, are the member nodes that need not to be hidden from others.

(3) *Middle Outsider*, **MO** nodes in short, are the nodes that do not need to receive any packets from the source for their own purpose, but providing packet forwarding service for the multicast system. If needed MO nodes are invited by the system for improving the overall efficiency. They do not hide their identity.

One naïve approach to achieve anonymous multicast services is to treat multicast as a set of unicast communications from the sender to the individual receivers and then directly apply one of the unicast anonymity schemes discussed in Section 2. While the approach is simple, it is inefficient. To achieve high efficiency and reduce the redundancy of multicast message transmissions among multiple receivers, multicast always relies on some structure present to deliver a message. The structure is usually a tree, and the tree can be source-based or core-based. Unicast is a special case of multicast, where the structure is a path. The potential solution to anonymous multicast must center on the concept of the tree. We believe it is the main difference and also the source of challenges in achieving anonymity in multicast compared with anonymity in unicast.

3.2 Design Consideration of Anonymous Multicast Systems

We need to consider both multicast tree efficiency and anonymity degree in the design of a multicast mutual anonymity protocol. An example is shown in Figure 1, in which we can see that an optimal multicast tree without (Figure 1(a)) and with (Figure 1(c)) anonymity concern are very different, where the cost of an AM-NM connection is 6 times that of an NM-NM connection and the cost of an AM-AM connection is 15 times that of an NM-NM connection. We have the following objectives in designing MAM protocol.

(1) High mutual anonymity degree: the identity of each anonymous node (AM), whether a sender or a receiver, in a multicast group should be hidden from all group members and outsiders.

(2) Delivery efficiency: a smart tree with consideration of anonymity is built with low average delay and low resource usage.

(3) Distributed fashion: the construction of the anonymous multicast system must be completely self-organizing and in a distributed manner. No trusted central server is involved. Further, MAM must be robust in a dynamic overlay environment.

(4) Self-optimization: MAM will allow all the nodes to incrementally optimize the system, by reconstructing the tree and inviting more middle outsiders (MO nodes) to improve the overall performance.
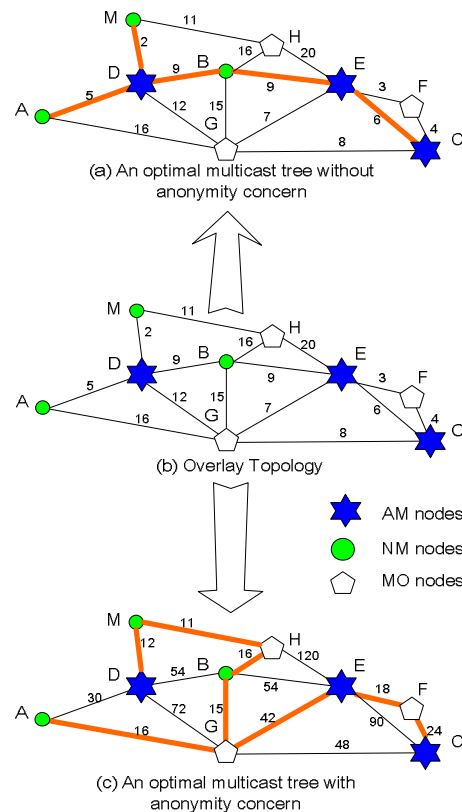

Figure 1: An example of multicast tree with and without anonymity concern

Here is the basic idea of MAM. A set of NM nodes form an efficient multicast tree in terms of bandwidth and/or delay. Nodes of the tree are degree-bounded. Early AM nodes connect unsaturated NM nodes, or MO nodes (if MO nodes have been invited) on the tree using a unicast initiator anonymity protocol. When there is no unsaturated NM node in the tree, a joining AM node will connect to another unsaturated AM node in the tree using a unicast mutual anonymity protocol. If there are too many AM nodes in the system, MO nodes will be invited to join the multicast tree so that the new AM node can connect with the MO node using a unicast anonymity protocol. When to invite MO nodes depends on the cost ratio of unicast initiator anonymity protocol and unicast mutual anonymity protocol, and the ratio of AM nodes in a system. The pseudo code for a joining node P is:

P contacts a bootstrapping server and gets a list of members;
P contacts one active member and gets a full list of members;
If (P is an NM node)
   make a direct connection to an unsaturated NM node;
If (P is an AM node) {
   If (P can find unsaturated NM members)
      make AM-NM connections with few unsaturated NMs;
   else
      If (P can find unsaturated AM members)
         make AM-AM connections with several AMs;
If (# of AMs/# of NMs > IT)
   Invite MO nodes;
If (timeout)
   tree optimization;

Therefore, there are mainly the following three key issues to be addressed in MAM: a unicast initiator anonymity protocol for AM-NM connections, a unicast mutual anonymity protocol for AM-AM connections, and anonymous multicast tree construction and optimization, which are discussed in detail in following subsections.

3.3 Unicast Initiator Anonymity Protocol Design

The idea of Onion and a reverse Onion can be used to achieve initiator anonymity for bi-directional communication. Since an AM node has more than one choice for NM nodes to make AM-NM connections with, we optimize the Onion protocol as follows to keep both strong anonymity and robustness.

In the improved protocol for AM-NM connection, a *Remailer* (a reverse Onion) is generated by the AM node for the NM node to anonymously send messages to the AM node. In the AM→NM communication direction, the AM node uses an approach similar to Crowds and Tor, in which each middle nodes in the path can make a decision to forward the message to another middle node or the NM node. This approach is more robust than Onion in the case of middle node failure. In order to simplify the protocol description, we use S to denote the AM node, and use R to denote the NM node, as below. Note that S knows R's identity, but R does not know anything about S. Since this connection is initiated by S, we also label S as the initiator. In the rest of the paper, we use where $\{M\}K$ indicates that M is encrypted with the key K. Kp_+ denotes p's public key and Kp_- denotes p's private key.

Step 1: The node S first generates m, the number of middle nodes in the *Remailer*. S then randomly selects a list of m nodes, $p_0, p_1, p_2, \ldots, p_{m-1}$ to form a *Remailer*. The lifetime of this one-time *Remailer* in seconds is also generated. The *Remailer* is built with S as the last member of the path and with p_i in the middle. It is of the form:

$\{p_{m-1}, \{p_{(m-2)}, \ldots \{p_0, \{S\}Kp_{0+}\ldots\}Kp_{(m-2)+}\}Kp_{(m-1)+}\}K_{R+}$.

Step 2: S randomly selects a node, q_0, sends it the message: $S \rightarrow q_0$: $\{R, \{Remailer, \text{lifetime}\}K_{R+}\}$.

Step 3: A peer q_i can elect itself to act as a *deliver* with a predefined forwarding probability h. If q_i is self-elected, the message $\{Remailer, \text{lifetime}\}K_{R+}$ will be delivered to the non-anonymous member node R directly. Otherwise, q_i will randomly select another node, q_{i+1} and forward the message $\{R, \{Remailer, \text{lifetime}\}K_{R+}\}$ to it.

Step 4: On receiving the message $\{Remailer, \text{lifetime}\}K_{R+}$, R uses its private key to decrypt the encrypted message.

Step 5: R generates a symmetric key K, and encrypts the multicast packet f with K. R then encrypts K with its private key. It keeps sending multicast packets with the format as below through the *Remailer* to S:

R→S: $\{f\}K, \{K\}K_{R-}$

Step 6: S uses R's public key to decrypt the symmetric key K, and uses K to decrypt the content encrypted by K.

At any time, a node R may have one or more *Remailers*. It will check the age and the expected lifetime for each *Remailer* periodically and delete obsolete *Remailers*. Each live *Remailer* corresponds to one AM node. Each AM node may connect with different NM nodes with the same or different *Remailers* for two reasons: increasing the difficulty for a NM node to guess the identity of the AM node, and providing multiple paths to the AM node in case of failure of any middle nodes in a *Remailer*. Two many *Remailers* for the same AM node will increase overhead, which can be adjusted by setting shorter lifetimes for the *Remailers*.

3.4 A Unicast Mutual Anonymity Protocol Design

When a joining AM node cannot find an unsaturated NM node in the tree, one option for him is to connect to another unsaturated AM node in the tree using a unicast mutual anonymity protocol. Most unicast mutual anonymity protocols [12, 23, 26, 28, 32] were proposed for file sharing systems and may not be applicable here directly because of their low efficiency. Therefore, we need to design a new unicast mutual anonymity protocol. Designing an efficient mutual anonymity protocol is difficult, but is possible here by utilizing the mechanism of MO node invitation.

We can use IP addresses to identify NM/MO nodes because they do not need to be anonymous. Instead, each AM node randomly selects an 18 byte value using a certain algorithm to ensure its uniqueness when it joins the system. Note that AM nodes may change their ID_{AM} at any time for anonymity consideration. At the time of its joining, each AM node is bounded with one or multiple MO nodes, which means an AM node sends *Remailers* to its bounded MO nodes, and its ID_{AM} and bounded MO nodes' IP addresses, e.g. ID_{AM}-IP_{MO1}, …, ID_{AM}-IP_{MOi}, …, will be kept in other NM nodes.

When an AM node (AM1) decides to make a connection to another AM node (AM2), it will select one of its bounded MO nodes (MOi) to establish a connection with and one of AM2's bounded nodes (MOj). The connection between AM1 and MOi, and the connection between AM2 and MOj are established by the unicast initiator anonymous protocol introduced in the previous section. A connection of ID_{AM1}-IP_{MOi}-IP_{MOj}-ID_{AM2} is therefore established to achieve mutual anonymity between AM1 and AM2.

3.5 Anonymous Multicast Tree Construction

Many previous studies have intensively studied how to build an efficient overlay and optimize a random overlay, so we will not focus on this issue in this paper. We use an idea similar to the Narada protocol [10] to build our multicast overlay among NM/MO nodes. The basic idea of Narada is to construct an efficient connected mesh first. Narada then constructs shortest path spanning trees of the mesh, each tree rooted at the corresponding source using well known routing algorithms.

As in Narada, every NM node and invited MO node has a full list of all the members. A joining node is able to get a list of group members (not necessary complete or accurate) by an out-of-band bootstrap mechanism, and randomly selects several unsaturated members to connect with. If the new joining node is an AM node that needs to hide its identity, it will randomly select one or multiple unsaturated NM/MO nodes forming anonymous connections with them, which is described in Section 3.3.

The multicast tree needs to be maintained. All the NM nodes probe their distances with all the other NM nodes and share the information among the overlay, so that every single node has an identical distance table including each pair of the NM nodes. As our design is for small sized systems, maintaining such a list is not difficult. With such a table, a good multicast tree including all NM nodes can be easily computed and maintained [10]. The distance between a NM node and an AM node is not available because the AM node is anonymous to NM nodes, but it is also not necessary since the AM node is connected with a NM node via a number of middle nodes making the direct distance between the AM node and the NM node meaningless in optimizing the tree. However, the ID_{AM}s of the AM nodes

can be kept in the NM nodes, and a NM node knows the number of AM nodes that connect with it via anonymous passage but does not know their identities. In optimizing the tree, this NM node will subtract the number of its connected AM nodes from its bounded degree.

When a joining AM node cannot find an unsaturated NM node in the tree, one option for him is to connect to another unsaturated AM node in the tree using a unicast mutual anonymity protocol described in Section 3.4. However, we do not wish to see too many AM-AM mutually anonymous connections for performance reasons. Therefore, in some situations, MAM considers inviting some MO nodes to help by joining the system. We define an Invitation Threshold, IT. When the ratio of AM nodes to NM/MO is greater than the value of IT, the system will try to invite some MO nodes to join. When MO nodes are invited into the systems, joining AM nodes will have chances to join the tree by making AM-NM connections instead of more expensive AM-AM connections.

3.6 Cost and Latency of Anonymous Connections

There is additional cost and latency for multicast systems when we try to provide anonymity to a set of member nodes, and hence it is of great importance to discuss this cost and latency. We have the following observations on the cost and latency of the above proposed unicast mutual anonymity protocols of MAM.

First, the selection of the number of middle nodes, m, has great impact on the anonymity degree and the cost of the connections. Obviously there is a tradeoff between the anonymity degree and the cost. Specifically, a larger m will provide a higher anonymity degree while incurring larger cost and latency.

Second, the predefined forwarding probability h also partially influences the cost and latency of data delivery in the system. In MAM, for simplicity, we uniformly select the value of h for the peering nodes. In real systems, nodes may select h independently, and the variety of h will improve the anonymity degree provided to the clients.

Third, the average cost of an AM-AM connection is at least two times greater than an AM-NM/MO connection. If we take (1) the dynamic nature of the member nodes, and (2) each AM node may use a set of NM/MO and switch some of them, into consideration, the average cost of an AM-AM connection is more than twice of that of an AM-NM/MO connection.

4. Anonymity Degree Analysis

4.1 Attack Model

We assume the attacker will break into some overlay nodes chosen randomly in one round and try to figure out who the AM node is using the information he gets from some broken nodes. We assume the attacker can find the

single parent and k children of all the nodes that have been broken. We also assume that the broken node keeps forwarding the packets in the same way as before it is broken. We call the parents of all those broken nodes the potential root of a subtree with AM nodes, which is called an implicit tree. The attacker will give each potential root a coefficient that is related with the probability regarded by the attacker as the root of the implicit tree by utilizing the information he gets from all the broken nodes. A node that is more likely to be the root of the implicit tree has a higher coefficient, which means it is more important than other broken nodes and more prone to further attack.

The objective of the attacker is to use the above coefficients for future attacks, e.g., the attacker can launch congestion attack to the potential root(s) to deny the service of as many receivers as possible, or the attacker can launch another break in attack to the potential root(s) to find the identity of the root of the implicit tree. No matter what the next attack is, the attacker will try to attack the node(s) that are more likely to be closer to the root of the implicit tree since he can potentially deny service to more receivers if he launches a congestion attack or has a higher probability to get the identities of all the receivers in the implicit tree if he launches a break in attack.

One thing to be reminded of here is that two broken nodes that are two layers apart can generate a broken tree with length of three by sharing information with each other. An example is that node A is in the ith layer, while node B is in the $(i+2)^{th}$ layer. After sharing the parent and children information with each other, node A finds that the parent of node B is actually one of its children, so a three layer broken tree is generated by nodes A and B. In this case, an unbroken node can also be on a broken tree as long as both its parent and at least one of its children are broken. We call node A the head of the broken tree if and only if node A is broken while its parent and grandparent are not broken. Similarly, we call node B the tail of the broken tree if and only if node B is broken while none of its children and grandchildren are broken. If node A is in the ith layer and node B is in the jth layer, we call this broken tree a broken tree with the length j-i+1, which is basically the number of nodes in this broken path. Generally, all the broken nodes can form a broken forest comprised of several broken trees that are subtrees of the implicit tree. We denote the length of a broken tree as the length of the longest broken path in the broken tree.

4.2 Anonymity Degree Analysis

The metric we use to analyze anonymity degree is P_{reveal}, which is the probability that the identity of an AM node is revealed. If the AM node itself is broken, this probability is 1; otherwise, we calculate this probability according to a weight. Each node has a weight that stands for how sure the attacker thinks that this node's parent or one of its children

is an AM node. Each node could be the root of a broken tree or the tail of a broken path, which we will define later. We believe the longer the broken tree or the broken path is, the more weight the attacker will give to this node.

We assume here the multicast tree structure is a k-nary incomplete tree with L+1 layers and the root node is at Layer 0. Here an incomplete tree means that some receivers are not in the Lth layer. The receivers can be located from the first layer to the Lth layer. We assume in the incomplete tree scenario, each node has either 0 or k children. We introduce the incomplete tree in the hope of achieving better bandwidth efficiency since there is no redundant link in an incomplete tree. Here, we introduce a set of parameters $\{q_{i,j}\}$, which is a value given to each node $p_{i,j}$ in the tree. We let $q_{i,j}$ be 1 if node $p_{i,j}$ is a real node in the tree. We let $q_{i,j}$ be 0 if it does not exist in the tree. We also assume that the attacker has successfully broken into N nodes in this tree. Since the attacker chooses the nodes randomly for break in attack, the probability of each node in the tree being broken is equal, which is shown below.

$$P_{broken} = N / \sum_{i=0}^{L} \sum_{j=1}^{k^i} q_{i,j} \tag{1}$$

If the root of the tree is one of the broken nodes, the attacker has already obtained all the information he needs. Otherwise, there is a probability p_{attack}^s that the real root will be regarded as the root and may be subject to the next attack. The overall probability that the identity of the root is revealed is,

$$P_{reveal}^s = P_{broken} + (1 - P_{broken}) * P_{attack}^s \tag{2}$$

$$p_{attack}^s = \sum_{j=1}^{k} (w_{1,j}^s / \sum_{i=1}^{L} \sum_{j=1}^{k^i} w_{i,j}^s) \tag{3}$$

$$w_{i,j}^s = \begin{cases} 0 & (q_{i,j} = 0) \\ P_{broken} * (\sum_{l=1}^{L} p_{i,j}^s(l) * f(l)) & (q_{i,j} = 1, i = 1) \\ P_{broken} * (1 - P_{broken}) * (\sum_{l=1}^{L} p_{i,j}^s(l) * f(l)) & (q_{i,j} = 1, i = 2) \\ P_{broken} * (1 - P_{broken})^2 * (\sum_{l=1}^{L} p_{i,j}^s(l) * f(l)) & (q_{i,j} = 1, i > 2) \end{cases} \tag{4}$$

$w_{i,j}^s$ is the weight given to node $n_{i,j}$ (i.e. j^{th} node in i^{th} layer), which is the head of a broken tree. $p_{i,j}^s(l)$ is the probability that the length of the broken tree with node $n_{i,j}$ as the head is l. $f(l)$ is a function that increases when l increases. The exact form of $f(l)$ depends on the attacker's policy. We choose $f(1)=1$ in this paper.

In an incomplete tree, different nodes at the same layer have different probabilities of being the head of a broken tree of a specific length. A node that has more "deeper"

descendant has higher probability of being a head of a long broken tree and vice versa.

$$
p_{i,j}^{s}(l) = \begin{cases}
0 & (q_{i,j} = 0) \\
0 & (l \leq 0 \, or \, l > L) \\
1 & (q_{i+1,kj} = 0, l = 1) \\
(1-p_{broken})^{k+\sum_{n=1}^{l} q_{i+2,k^2 j+\cdots}} & (q_{i+1,kj} = 1, l = 1) \\
0 & (q_{i+1,kj} = 0, l > 1) \\
\sum_{j=1}^{\sum_{n=1}^{l} q_{i+2,k^2 j+\cdots}} (p_{bg}(j) * p_{i+2,j}^{s}(l-2 \mid bn = j)) \\
\quad * p_{bc}(0) + \sum_{j=1}^{k} (p_{bc}(j) * p_{i+1,j}^{s}(l-1 \mid bn = j)) & (otherwise)
\end{cases} \quad (5)
$$

Here, $p_{bc}(i)$ is the probability that i children have been broken. Similarly, $p_{bj}(i)$ is the probability that i grandchildren have been broken. $p_i(l \mid bn = 1)$ is the probability that the longest broken tree among j trees is of length l, given that j children of a parent, which is in the $(i-1)^{th}$ layer, have been broken in the ith layer. $p_{bc}(i)$, $p_{bj}(i)$ and $p_i(l \mid bn = 1)$ can be calculated as:

$$
p_{bc}(i) = \binom{k}{i} * p_{broken}^{i} * (1-p_{broken})^{k-i} \quad (6)
$$

$$
p_{bg}(i) = \binom{\sum_{n=1}^{k^2} q_{i+2,k^2 j-k^2+n}}{i} * p_{broken}^{i} * (1-p_{broken})^{\sum_{n=1}^{k^2} q_{i+2,k^2 j+\cdots} - i} \quad (7)
$$

$$
p_{i,j}^{s}(l \mid bn = j) = \begin{cases}
0 & (l = 0) \\
\sum_{m=1}^{j} \left(\binom{j}{m} * \overline{p_{i,j}^{s}(l)}^{m} \right) \\
\quad * (\sum_{n=1}^{l-1} \overline{p_{i,j}^{s}(n)})^{j-m}) & (l > 0)
\end{cases} \quad (8)
$$

Here, $\overline{p_{i,j}^{s}(l)}$ can be calculated as below,

$$
\overline{p_{i,j}^{s}(l)} = (\sum_{j=k*i-k+1}^{k*i} p_{i,j}^{s}(l)) / k \quad (9)
$$

So far, we have finished analyzing how to get p_{reveal}.

For anonymity degree of AM as a receiver, we denote the AM node we consider as $p_{u,t}$. Its parent is $p_{u-1,[t/k]}$ and grandparent is $p_{u-2,[[t/k]/k]}$. Here, we denote [i] as the largest integer that is no more than i. We give the formulae to calculate the probability that the identity of the AM as a receiver is revealed as below.

$$
P_{reveal}^{r} = (1 - (1 - P_{broken})^{2}) + (1 - P_{broken})^{2} * P_{attack}^{r} \quad (10)
$$

$$
w_{i,j}^{r} = \begin{cases}
0 & (q_{i,j} = 0) \\
0 & (q_{i,j} = 1, q_{i+1,k*j} = 0) \\
P_{broken} * (1 - P_{broken}) \\
\quad *(\sum_{l=1}^{L-1} p_{i,j}^{r}(l) * f(l)) & (q_{i,j} = q_{i+1,k*j} = 1, q_{i+2,k^2*j} \\
& = 0, \{i,j\} \neq \{u-1,[t/k]\}) \\
(\sum_{l=1}^{L-1} p_{i,j}^{r}(l) * f(l)) & (q_{i,j} = q_{i+1,k*j} = 1, q_{i+2,k^2*j} \\
\quad * P_{broken} & = 0, \{i,j\} = \{u-1,[t/k]\}) \\
P_{broken} * (1 - P_{broken})^{2} & (q_{i,j} = q_{i+1,k*j} = q_{i+2,k^2*j} = 1, \\
\quad *(\sum_{l=1}^{L-1} p_{i,j}^{r}(l) * f(l)) & \{i,j\} \neq \{u-2,[[t/k]/k]\}) \\
P_{broken} * (1 - P_{broken}) & (q_{i,j} = q_{i+1,k*j} = q_{i+2,k^2*j} = 1, \\
\quad *(\sum_{l=1}^{L-1} p_{i,j}^{r}(l) * f(l)) & \{i,j\} = \{u-2,[[t/k]/k]\})
\end{cases} \quad (11)
$$

$$
p_{attack}^{r} = \begin{cases}
0 & u = 1 \\
w_{u-1,[t/k]}^{r} / k & u > 1 \\
\overline{\sum_{i=1}^{L} \sum_{j=1}^{k^i} w_{i,j}^{r}}
\end{cases} \quad (12)
$$

Here, the definition of $p_{i,j}^{r}(l)$ is similar to $p_{i,j}^{s}(l)$, which is defined before.

$$
p_{i,j}^{r}(l) = \begin{cases}
0 & (l > u - 1) \\
0 & (l < 1 \, or \, i < 1) \\
1 & (l = 1, i = 1) \\
1 - p_{broken} & (l = 1, i = 2) \\
(1 - p_{broken})^{2} & (l = 1, i > 2) \\
p_{broken} * p_{i-1,[j/k]}^{r}(l-1) + (1 - p_{broken}) \\
\quad * p_{broken} * p_{i-2,[[j/k]/k]}^{r}(l-2) & (otherwise)
\end{cases} \quad (13)
$$

4.3 Numerical Results and Discussions

In the following discussion, we will consider the numerical results based on the above formulae for anonymity degree in the incomplete tree. The incomplete tree that we use is a binary tree. The root node has four grandchildren: one is the root of a complete subtree with 2 leaves in the third layer, one is the root of a complete subtree with 4 leaves in the fourth layer, one is the root of a complete subtree with 8 leaves in the fifth layer, and the other is the root of a complete subtree with 16 leaves in the sixth layer. The data are obtained in MATLAB.

Figure 2 and 3 show the sensitivity of anonymity degree to broken ratio. Different curves represent different combinations of k and L. Anonymity degree is represented by P_{reveal}. Smaller P_{reveal} results in better anonymity degree. It is obvious that anonymity degree improves as broken ratio decreases.

When the percentage of broken nodes and L are fixed, anonymity degree improves when k increases. This is because when the tree grows wider, the broken nodes tend to be in different branches. The length of the broken tree
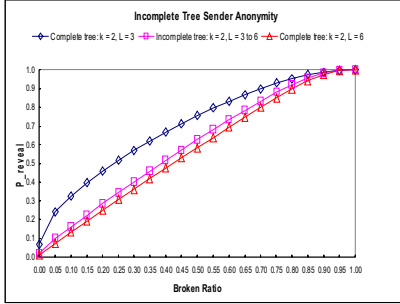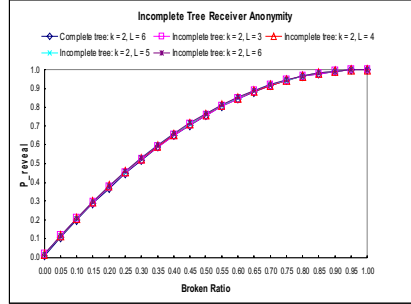
Figure 2: Ananymity degree of AM as a sender

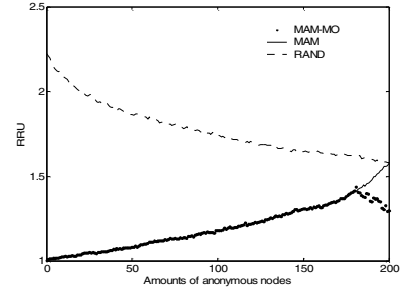**Figure 3: Ananymity Degree of AM as a receiver**

Figure 4: RRU vs the number of AM nodes

tends to decrease. When the percentage of broken nodes and k are fixed, anonymity degree improves when L increase. This is because when the tree grows deeper, the length of the broken tree tends to decrease.

The AM sender anonymity of the incomplete tree in our example in Figure 2 is between those of the complete binary tree with all receivers in the third or sixth layer. This is obvious because the AM sender's anonymity improves when the tree grows. We observe that the difference between the incomplete tree curve and the complete tree curve with six layers is very slight. This is because the children of the sender who has fewer descendants will have comparatively small weight, which helps to improve the AM sender's anonymity. Actually we can achieve significant bandwidth efficiency with little sacrifice on AM sender's anonymity.

The AM receiver anonymity of the incomplete tree in our example in Figure 3 is worse than that of the complete tree with sixth layers. This is because fewer nodes will be considered as the parent of the receiver, the comparative weight of the AM receiver's parent tends to increase. We observe that among all the AM receivers in the incomplete tree, the higher AM receivers have better anonymity than the lower ones. This is because their parents tend to be the tail of a shorter broken tree, which helps to decrease their weight. This fact holds under the assumption that the attacker does not know the layer of the AM receiver. We observe that the differences among different AM receivers in the incomplete tree case and between the incomplete tree case and the complete tree case are very slight because the AM receiver's anonymity is dominated by the probability that the sender or the receiver is broken, which is determined by the percentage of broken nodes. This means that significant bandwidth efficiency can be achieved with little sacrifice of AM receiver's anonymity.

5. Performance Evaluation

5.1 Simulation Methodology

Two types of topologies, physical and logical topologies, are generated in our simulation. The physical topology should represent the real topology with Internet character-

istics. The logical topology represents the overlay system built on top of the physical topology. To simulate the MAM protocol in a more realistic environment, both topologies must accurately reflect the topological properties of real networks in each layer. BRITE [1] is a topology generation tool that provides the option to generate topologies based on the AS Model. Using BRITE, we generate physical topologies with 3,000 to 7,000 nodes. The average number of neighbors of each node ranges from 4 to 10. The 100 to 300 overlay nodes are randomly selected from the nodes in the physical topologies.

To reflect the real overlay systems, in the experiments we report here, member nodes are coming and leaving according to the distribution observed in [25]. The mean of the distribution is chosen to be 1800 seconds. The value of the variance is chosen to be half of the value of the mean. In each experiment, a number of nodes join the system at the first 120 seconds of the simulation in random sequence. The lifetime of each node will be decreased by one after passing each second. A member will leave in the next second when its lifetime reaches zero. During each second, there are a number of members leaving the system, and we then randomly pick up (turn on) a similar number of members from the physical network to join the system.

In all the experiments, every 50 seconds, random nodes are selected as senders to multicast data at a constant rate, and the simulations run for 60 minutes. In the MAM protocol, the lifetime of *Remailers* is randomly selected from 50 to 200 seconds.

5.2 Performance Metrics

We compare the performance of three different approaches: Optimal, MAM, and RAND. In Optimal, the anonymous multicast tree is optimized using an offline algorithm. In a naïve approach, indicated as "RAND", each joining node randomly selects a member to connect to the multicast tree.

We use two performance metrics: *relative resource usage* (RRU) and *average worst-case delay* (AWD).

The stress of a physical link is defined in [10] as the number of identical copies of a packet carried by a physical

link. We define resource usage as $\sum_{j=1}^{N} d_j \times s_j$, where d_j is the delay of link j and s_j is the stress of link j. Resource usage is one of the parameters of seriously concerned to network administrators. Heavy network traffic limits the scalability of overlay networks [24]. RRU is defined as the ratio of the resource usage of MAM or other approaches to the optimal anonymous multicast tree. AWD is the average delay from the source to the farthest node that gets the multicast packets, when nodes are selected at random as the source nodes in multiple runs.

5.3 Simulation Results

When there is no member needing to hide its identity, then the system will be the same as normal end system multicast. Intuitively, when more nodes need to be hidden, the total cost of the system will increase. We first show MAM's performance by increasing the number of nodes that need to achieve anonymity (AM nodes) in the system.

With 3000 physical nodes and 200 overlay multicast members, Figures 4 and 5 plot the RRU and AWD of different approaches versus the number of AM nodes in the system. When the ratio of AM nodes in the system is small, MAM's RRU is very close to the optimal solution. MAM's AWD is very close to the optimal solution when less than half of the nodes are AM nodes. We vary the system size from 100 to 400, and the physical network size from 2,000 to 8,000. The results are consistent, indicating that MAM maintains effectiveness, and the RRU and AWD of MAM are not sensitive to the system size or the physical networks size. If all of the members in a system are AM nodes, even the optimal solution is as bad as the naïve RAND approach, and a system of smaller size could incur greater traffic overhead than a system with larger size. Hence, in MAM, we propose to avoid having all the members as AM nodes by inviting MO nodes into the system. Frankly, it is always helpful if more MO nodes can join the system. However, the overhead of inviting MO nodes is hard to predict: they merely provide service to the system but do not consume the multicast content.

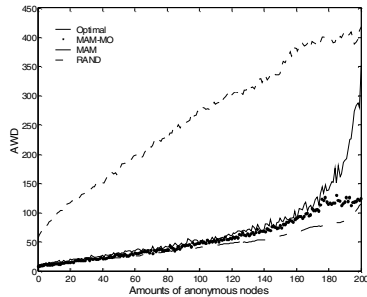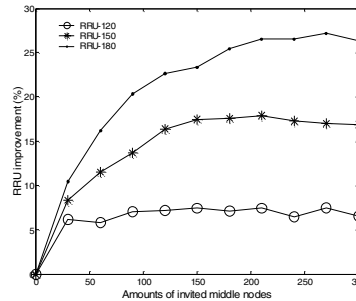In the "MAM" protocol, MO nodes are not invited. We can see that when the percentage of AM nodes is large, both RRU and AWD degrade significantly. We investigate the effectiveness of inviting MO nodes to join in Figures 4 and 5 using the "MAM-MO" protocol, in which MO nodes are invited when the ratio of AM nodes in the system reaches 90%. The improvement is substantial for a system with more than 270 AM nodes (90% of the system).
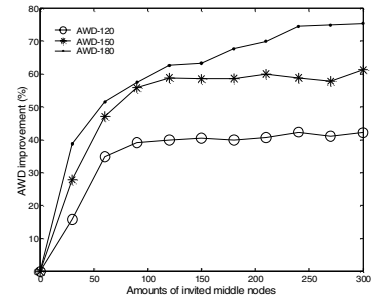
The next question is when is the best time for the system to invite MO nodes, i.e. what is the best Invitation Threshold IT.

Figures 6 and 7 show the RRU improvement and AWD improvement versus the number of invited MO nodes for a different given number of AM nodes in a system with 200 overlay multicast members. RRU/AWD improvement is defined as the percentage of the RRU/AWD improvement with the MAM-MO protocol over the MAM protocol without MO node invitation. "RRU-n"/ "AWD-n" means the RRU/AWD improvement for a given number of n AM nodes. In general, inviting more MO nodes means better performance with the assumption that we have an infinite number of available MO nodes to be invited.

However, when a certain number of MO nodes have been invited, inviting more MO nodes is not as effective as before. For example, there is a clear jump in Figure 6 for RRU-120, which shows that when 30 MO nodes have been invited, inviting more MO nodes gives little additional RRU improvement, where the corresponding IT is 120/(200+30)=52%. Similarly, the ITs for RRU-150 and RRU-180 are 47% and 47%. If we calculate the ITs from Figure 7, we have 46%, 52% and 47% for AWD-120, AWD-150, and AWD-180 respectively.

Therefore, our interpretation of the experimental results is that when less than around 50% of the nodes wish to be anonymous, MAM may be directly used with no need to invite MO nodes; otherwise, MO nodes should be invited to keep the ratio of AM nodes in the system at about 50%. Beyond this, inviting more MO nodes is not necessary.

6. Conclusion and Future Work

In this paper, we propose the MAM protocol to provide anonymous multicast service. Our analysis shows that the

Figure 5: AWD vs the number of AM nodes

Figure 6: RRU improvement vs. # of invited MO nodes

Figure 7: AWD improvement vs. # of invited MO nodes

anonymity degree of AM nodes is correlated with the broken ratio, tree degree, and tree depth. We also show that the incomplete multicast tree can achieve a similar anonymity degree with much higher bandwidth efficiency, compared to the complete multicast tree.

Our performance evaluation shows that MAM is an effective approach to constructing an efficient anonymous multicast tree. When the percentage of AM nodes in a system is below a certain level, without inviting MO nodes, MAM works as well as the optimal solution. We have also show that inviting a certain ratio of MO nodes can be very effective for a system with a large number of AM nodes.

REFERENCES

[1] BRITE, http://www.cs.bu.edu/brite/

[2] "RSAREF20, http://tirnanog.ls.fi.upm.es/Servicios/Software/ap_crypt/disk3/rsaref20.zip," 1994.

[3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," *Proceedings of ACM SIGCOMM*, 2002.

[4] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M.Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," *Proceedings of INFOCOM*, 1999.

[5] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE JSAC*, 2002.

[6] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An Evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays," *Proceedings of IEEE INFOCOM*, 2003.

[7] D. Chaum, "Untraceable Electronic Mail Return Addresses, and Digital Pseudonyms," *Communications of the ACM, pp.84-88*, 1981.

[8] Y. Chu, A. Ganjam, T. Ng, S. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early Experience with an Internet Broadcast System Based on Overlay Multicast," *Proceedings of USENIX Annual Technical Conference*, 2004.

[9] Y. Chu, S. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture," *Proceedings of ACM SIGCOMM*, 2001.

[10] Y. Chu, S. G. Rao, and H. Zhang, "A Case for End System Multicast," *Proceedings of ACM SIGMETRICS*, 2000.

[11] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," *Proceedings of 13th USENIX Security Symposium*, 2004.

[12] M. Freedman and R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer," *Proceedings of CCS*, 2002.

[13] X. Fu, B. Graham, D. Xuan, R. Bettati, and W. Zhao, "Analytical and Empirical Analysis of Countermeasures to Traffic Analysis Attacks," *Proceedings of IEEE International Conference on Parallel Processing (ICPP)*, 2003.

[14] E. Gabber, P. Gibbons, D. Kristol, Y. Matias, and A. Mayer, "Consistent, Yet Anonymous,Web Access with LPWA," *Communications of the ACM, 42 2. pp.42-47, February*, 1999.

[15] E. Gabber, P. Gibbons, Y. Matias, and A. Mayer, "How to Make Personalized Web Browsing Simple, Secure, and Anonymous," *Proceedings of Conference on Financial Cryptography*, 1997.

[16] C. Grosch, "Framework for Anonymity in IP-Multicast Environment," *Proceedings of IEEE GLOBECOM*, 2000.

[17] Y. Guan, X. Fu, D. Xuan, P. Shenoy, R. Bettati, and W. Zhao, "NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications," *IEEE Transactions on Systems, Man, and Cybernetics*, 2001.

[18] I. Gupta and K. Bitman, "Holistic Operations in Large-scale Sensor Network Systems: a probabilistic peer-to-peer approach," *Proceedings of International Workshop on Future Directions in Distributed Computing (FuDiCo)*, 2002.

[19] P. Kruus and J. Macker, "Techniques and issues in multicast security," *Proceedings of MILCOM*, 1998.

[20] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni, "A Distributed Approach to Solving Overlay Mismatch Problem," *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS)*, 2004.

[21] M. Moyer, J. Rao, and P. Rohatgi, "A survey of security issues in multicast communications," *IEEE Network*, 1999.

[22] A. Nakao, L. Peterson, and A. Bavier, "A Routing Underlay for Overlay Networks," *Proceedings of ACM SIGCOMM*, 2003.

[23] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Transactions on Information and System Security, pp. 66-92, November*, 1998.

[24] Ritter, Why Gnutella Can't Scale. No, Really, http://www.tch.org/gnutella.html

[25] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Proceedings of Multimedia Computing and Networking (MMCN)*, 2002.

[26] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "P5: A Protocol for Scalable Anonymous Communication," *Proceedings of IEEE Symposium on Security and Privacy*, 2002.

[27] S. Shi and J. S. Turner, "Routing in Overlay Multicast Networks," *Proceedings of IEEE INFOCOM*, 2002.

[28] P. F. Syverson, D. M. Goldschlag, and M. G. Reed, "Anonymous Connections and Onion Routing," *IEEE Symposium on Security and Privacy (S&P'97), pp.44-53*, 1997.

[29] M. Waldvogel and R. Rinaldi, "Efficient Topology-aware Overlay Network," *Proceedings of ACM HotNets*, 2002.

[30] N. Weiler, "Secure Anonymous Group Infrastructure for Common and Future Internet Application," *Proceedings of In Proceedings of 17th Annual Computer Security Applications Conference (ACSAC'01)*, 2001.

[31] L. Xiao, A. Patil, Y. Liu, L. M. Ni, and A.-H. Esfahanian, "Prioritized Overlay Multicast in Ad-hoc Environments," *IEEE Computer Magazine, Page 67-74, February*, 2004.

[32] L. Xiao, Z. Xu, and X. Zhang, "Low-cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks," *IEEE Transactions on Parallel and Distributed Systems*, 2003.

[33] Z. Xu, C. Tang, and Z. Zhang, "Building Topology-aware Overlays Using Global Soft-state," *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, 2003.